REMARKS

Claims 10-11, 25-26 and 57-58 have been cancelled and new claims 78-79 have been added. Therefore, on entering this Amendment, claims 1-9, 12-24, 27-56, and 59-79 are all the claims pending in the application.

Claims 1-77 are pending in the application.

Claims 1-77 are rejected.

The drawings are objected to under 37 CFR 1.83(a).

Claims 1-43 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement.

Claims 3, 6, 24-26 and 56-58 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention.

Claims 1-10, 12-19, 21-26, 28-51, 53-58 and 60-77 are rejected under 35 U.S.C. 102(e) as being anticipated by Starr et al. U.S. Patent No. 6,807,581, (hereinafter Starr).

Claims 11, 20, 27, 52 and 59 are rejected under 35 U.S.C. 103(a) as being unpatentable over Starr as applied to claims 1, 22 and 23 above, and further in view of Muller et al. (U.S. Patent No. 6,453,360), (hereinafter Muller).

The Applicant traverses the rejections and requests reconsideration.

Drawings

The Examiner is believed to be incorrect in rejecting the drawings. Paragraph [0066] of the present Application clearly states "*The NI 220, which depending on the type of interface to be* *connected to as well as destination, routes the data in its network layer format through busses 222. Busses 222 may be Ethernet, ATM, or any other proprietary or standard networking interface.*" In Figure 2 busses 222 are comprised of busses 222-1 through 222-m and hence communication may be selected from a group consisting from those different types of busses, i.e., a bus 222-i conforming to a standard or proprietary communication protocol maybe selected.

Claim Rejections Under 35 USC § 112 – First Paragraph

Claim 1 is rejected by the Examiner as not being in compliance with 35 USC § 112, first paragraph. The Applicants respectfully disagree and draw the attention to Figs. 5A through 5I which show step by step how data is transferred without moving the data within the memory location. In related art, actual movement of data is required when data is transferred from the control of one device to another, as different memory locations are dedicated to different devices. Examiner is believed to be confusing the actual data transfer to and from a host and a networked device, and the transfers of data occurring **within a network interface card**.

In accordance with the disclosed invention an elaborate pointer system is used to avoid actual movement of data within a network interface card. This prevents the need to perform the intermediate step of moving data within a memory to signify the change of control of that data from one device to another. This capability is also described and supported in paragraph [0070] of the present Application. The Applicants amend the claims to clarify that there is no actual movement of data within memory locations in the data streamer.

Claims 2–43 are rejected because of their dependency on claim 1, and should therefore now be allowable.

Claim Rejections Under 35 USC § 112 – Second Paragraph

The claims have been amended to overcome the grounds for their rejections under section 112, second paragraph, and to place them in a better form.

Claim Rejections Under 35 USC § 102

Claims 1 – 10, 12 – 19, 21 – 26, 28 – 51, 53 – 58 and 60 – 77 are rejected as being anticipated by Starr et al. in US patent 6,807,581 (hereinafter "Starr").

Claims 10, 25, 26, and 57-58 are deleted, rendering their rejections moot.

As an initial matter, a discussion of at least two material differences between Starr and the present Application is provided. Specific differences between the claims and Starr are discussed subsequently.

General Comments – Data Movement.

A first difference has to do with the actual movement of data within the network interface card. According to Starr, data is moved within the INIC and evidence for that is found in multiple places. For example Starr notes that: "*Upon matching the packet summary with the CCB, assuming no exception conditions exist, the data of the packet, without network or transport layer headers, is sent by direct memory access (DMA) unit 68 to the destination in file cache 80 or file cache 24 denoted by the CCB.*" (column 8, lines 7-11). Further, it is noted earlier in Starr that "*When a network packet that is directed to the host 20 arrives at the INIC 22, the headers for that packet are processed by the sequencers 52 to validate the packet and create a summary or descriptor of the packet, with the summary prepended to the packet and stored in frame buffers 77 and a pointer to the packet stored in a queue.*" (column 6, lines 58-63). Hence, it is clear that Starr discloses data movement. In another example Starr notes that "*A packet control sequencer 510 oversees the fly-by sequencer 502, examines information from the media access controller 60, counts the bytes of data, generates addresses, moves status and manages the movement of data from the assembly register 500 to SRAM 508 and eventually DRAM 512. The packet control sequencer 510 manages a buffer in SRAM 508 via SRAM controller 515, and also indicates to a DRAM controller 518 when data needs to be moved from SRAM 508 to a buffer in DRAM 512. Once data movement for the packet has been completed and all the data has been moved to the buffer in DRAM 512, the packet control sequencer 510 will move the status that has been generated in the fly-by sequencer 502 out to the SRAM 508 and to the beginning of the DRAM 512 buffer to be prepended to the packet data.*" (column 13, line 55 through column 14, line 2).

By contrast, no such data movement is required in the present invention. This is specifically noted with the examples in Figs. 5A through 5I, and as further supported by the accompanying text of the present Application. Starr notes that the advantage of the architecture proposed therein is: "... *that the packet does not, in contrast with conventional sequential software protocol processing, have to be stored, moved, copied or pulled from storage for processing each protocol layer header, offering dramatic increases in processing efficiency and savings in processing time for each packet.*" (column 15, lines 33-39). Starr specifically uses the term storage which throughout Starr refers to a storage unit and not the memory used in Starr's INIC. Starr notes that the capability to terminate TCP collect the L5 (NFS in this case) data in the INIC's local DRAM, involves the copying of the data once, from local DRAM (Frame Buffer 77) to local DRAM (INIC File Cache 80), avoiding data copies by the host software.

According to the present Application it is shown that translation from TCP, layer 4, to Session layer, layer 5, is possible without locally copying the data. Hence, while Starr clearly provides an advantage over prior art of its time, it still does not resolve what is solved in the present Application, namely, eliminating the need to move data within the INIC itself. Therefore the techniques disclosed in the present Application is believed to overcome at least a performance limitation of Starr.

General Comments – Protocol Processing.

A second difference between Starr and the present Application has to do with the location of where protocol processing from layer 5 downwards is performed and controlled. In this respect the Applicants draw the Examiner's attention to Fig. 2 of Starr which clearly notes that protocol stack 38 is part of host 20. Starr therefore notes that "*Also stored in host memory 33 is a protocol stack 38 of instructions for processing of network communications and a INIC driver 39 that communicates between the INIC 22 and the protocol stack 38.*" (column 5, lines 18-21). Starr further notes that "*In order to provide fast-path capability at the host 20, a connection is first set up with the remote host, which may include handshake, authentication and other connection initialization procedures. A communication control block (CCB) is created by the protocol stack 38 during connection initialization procedures for connection-based messages, such as typified by TCP/IP or SPX/IPX protocols… When a message, such as a file write, that corresponds to the CCB is received by the INIC, a header portion of an initial packet of the* *message is sent to the host 20 to be processed by the CPU 30 and protocol stack 38. This header portion sent to the host contains a session layer header for the message, which is known to begin at a certain offset of the packet, and optionally contains some data from the packet. The processing of the session layer header by a session layer of protocol stack 38 identifies the data as belonging to the file and indicates the size of the message, which are used by the file system to determine whether to cache the message data in the host file cache 24 or INIC file cache 80, and to reserve a destination for the data in the selected file cache...*" (column 7, lines 23-65). Hence, it is clear that the protocol processing is performed under the control of host 20 and that for at least every initial packet of a message within an established connection there is a need to access host 20 for certain protocol processing; therefore, only partial relief is provided to that host. By contrast, and as can be seen in at least Figs. 4 and 6 as well as the supporting text in at least paragraphs [0072] and [0083] of the present Application, the protocol processing is performed at the data streamer level, providing a full relief from protocol processing by the host. Specifically it should be noted that upper level protocol (ULP) processing in performed by the data streamer, a capability not found in Starr's INIC. More specifically, in an established connection in accordance with the present Application there is no need to involve the host when a new message is received using the same connection. This of course not only relieves the host from processing load but also provides for a higher overall performance of the system. The differences are further noticeable from Figs. 12 and 13 of Starr that show the way that Starr communicates with the host to handle the upper layer protocol (ULP), and further in conjunction with the explanations provided by Starr in column 16 line 63 through column 17 line 36. Therefore in the present Application there is no interaction between the host and the data streamer during transfer of data, in contrast with Starr.

General Comments – Queues.

While queues are used in both Starr and the present Application, the mere fact that both exist does not constitute a reason to assume that these have an identical function. Starr has an elaborate scheme of queues having multiple functions. Starr refers to a plurality of queues (column 29, lines 12), and specifically mentions other queues, for example, a receive queue (column 14, line 4), a free buffer queue (column 29, line 41), vector queue 2113 (column 31, line 61), and summary queue 2112 (column 31, line 67).

By contrast, the present Application discloses only two queues, the object queue (520) and the application queue (530). For each connection, in accordance with the present Application, there is created an application queue is opened and managed until the connection is terminated. The operation of the object queue and an application queue is discussed in great detail with respect to Figs. 5A through 5I of the present Application and further in conjunction with descriptors 540. These queues are specifically designed to allow the operation of the present Application in such a way that in the interface between layer 4 and layer 5 processing there is no need to move data, a functionality explained in detail above, nor is there a need to access the host on a per massage basis. None of the queues suggested by Starr have any kind of functionality resemblance to the functionality suggested by the queues of the present Application, neither is there an equivalent to the per-connection application queue shown in detail in the present Application.

Claim 1. The Examiner attempts to read claim 1 onto features disclosed in Starr. The Applicants respectfully disagree that the data streamer is structured and functions in an equivalent way in Starr and the present invention. Examiner asserts that the data streamer of Starr is capable of transferring the data without moving the data within the memory location. Starr itself, provides evidence to the contrary as discussed in more detail in the general comments above. Specifically, Starr fails to disclose (or suggest) the present invention at least because it does not disclose (or suggest) that "data is transferred...without moving the data between memory location within the memory." The amendments to the claim further clarifies this point.

Independent claims 44, 76 and 77 include limitations analogous to claim 1. Therefore the arguments discussed above are analogously valid.

Claims 2 – 9, 12 – 19, 21 – 24, 28 –43, 45- 51, 53 – 56 and 60 – 75 are dependent on claims 1 or 44 and are allowable for the same reasons.

Further, regarding claim 3, even the most careful reading of column 5, lines 26-39 of Starr did not reveal that Starr provides for the use of the host solely for the initialization of the networked system disclosed in the present Application.

Regarding claim 12, the Examiner asserts that the "event queue manager" of the present Application is disclosed by Starr in column 24 line 59 through column 25 line 13. While a FIFO is described there in excruciating detail, Starr does not disclose that these FIFOs operate as an event queue discussed in the present Application, where it states that "*It should be further noted that the function of EQMS 260 is to control the operation of PNs 270.*" [0068]. Nothing of the sort is suggested by Starr. The Examiner asserts that the "control hub" (for example, CH 290 of the present Application) is equivalent with Starr elements 742 and 744 disclosed in column 21 lines 8-38. Reading of Starr reveals that "*... SRAM and DMA controller 740, which includes DMA controllers 742 and SRAM controller 744...*" (column 21, lines 21-22). In the present Application t*he function of CH 290 is to handle the control messages (as opposed to data messages)...*" (see for example, [0068]). The functions are clearly different to a person skilled in the art.

Regarding claim 22, the Examiner is believed to be incorrect in the equivalence that is drawn between the queues used for Starr (e.g., receive queue, receive buffer queue, receive statistics queue) and the queues used in the present Application. Specifically, the operation of the queues in accordance with the present Application and as further shown in conjunction with Figs. 5A-5I of the present Application, enable the operation of handling the packets without moving it as it gets processed, thereby overcoming a clear and acknowledged deficiency of Starr (see specifically flowcharts in Figs. 3 and 4 and descriptions thereof). The operation of the queues in Starr follow the standard approach and are not aimed to resolve the problem which is resolved by the present Application.

Regarding claims 28-29, the Examiner asserts that Starr in col. 30, line 53 through col. 31, line 9, teaches the disclosed invention. While pointers are used in both cases, the functionality is different. Starr's receive FIFO read pointer is used to cause the DRAM controller 742 to move data from the DRAM write data register to buffer 2114. This is of course in contradiction to all that is explained in detail in both the claims as well as the corresponding text that data is not to be moved during the time that data is in the networked system of the present Application. The Applicants respectfully submit that the operation of the two systems is not the same.

Claim Rejections Under 35 USC § 103

Examiner rejects claims 11, 20, 27, 52 and 59 as being unpatentable over Starr as applied to claims 1, 22 and 23 by the Examiner, and further in view of US patent 6,453,360 by Muller et al. (hereinafter "Muller"). Please note claim 11 is cancelled, rendering its rejection moot.

The above claims should be allowed for at least being a dependent claim of an allowable claim. Moreover, Muller does not overcome the deficiency noted in the teachings of Starr regarding not moving data within the memory of the data streamer.

Further, regarding claim 27, as mentioned above, the Examiner has failed to show that Starr suggests an object queue as used in the present invention. Specifically, data is not moved as control of the data moves from, for example, the host to the networked resources. This avoids unnecessary transfers of data from one memory location to the other. Nothing in Starr shows support of such mechanisms.

New Claims

New claims 78 and 79 are added for examination. These claims are supported by Fig. 2 and the accompanying description.

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such actions are hereby solicited. If any points remain in issue which the

Examiner feels may be best resolved through a personal or telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below.

The USPTO is directed and authorized to charge all required fees, except for the Issue Fee and the Publication Fee, to Deposit Account No. 19-4880. Please also credit any overpayments to said Deposit Account.

Respectfully submitted,

Chid S. Iyer
Registration No. 43,355

SUGHRUE MION, PLLC
Telephone: (202) 293-7060
Facsimile: (202) 293-7860

WASHINGTON OFFICE
23373
CUSTOMER NUMBER

Date: September 7, 2005